We claim:

1. An apparatus for providing a user interface to a designer of documents, said apparatus comprising:

Sub C1

an input for receiving input from said designer, said input comprising first event-driven programs;

an interpreter interpreting said event-driven programs into serial execution code;

an output for outputting said serial execution code to a server that serially executes said serial execution code,

upon command by said designer, said interpreter reinterpreting said serial execution code into second event-driven programs.

2. The apparatus of claim 1, further comprising:

a client connected to said server, said client receiving the output of said serial execution code;

wherein the user interface provided to said designer displays the programs that operate between said client and server as programs that operate as a single machine.

3. The apparatus of claim 1, wherein said event-driven programs include objects.

4. The apparatus of claim 3, further comprising: a script library for storing a script relating to objects for later placement in said first event-driven programs.

Sub B4

5. The apparatus of claim 1, said apparatus further comprising:

design-time controls for controlling the generation of said objects when said design-time controls are placed within said first event-driven programs.

Sub C2

6. The apparatus of claim 1, wherein said first and said second event driven programs are the same event-driven programs.

7. The apparatus of claim 1, wherein said first and second event driven programs are different event-driven programs.

8. A method for operating with a user interface provided to a designer of documents, said user interface representing documents as event-driven, said method comprising the steps of:

receiving an input from said designer, said input comprising first event-driven programs;

interpreting said event-driven programs into serial execution code;

outputting said serial execution code to a server that serially executes said serial execution code,

upon command by said designer, reinterpreting said serial execution code into second event-driven programs.

9. The method of claim 8, further comprising the step of:

receiving the output of said serial execution code at a client connected,

wherein the user interface provided to said designer displays the programs that operate between said client and server as programs that operate as a single machine.

10. The method of claim 8, wherein said event-driven programs include objects.

11. The method of claim 10, further comprising the step of:

storing in a script library a script relating to objects for later placement in said first event-driven programs.

sub B5 12. The method of claim 8, further comprising the steps of:

controlling the generation of said objects with controls that operate during a design time when said controls are placed within said first event-driven programs.

sub C 13. The method of claim 8, wherein said first and said second event driven programs are the same event-driven programs.

14. The method of claim 8, wherein said first and second event driven programs are different event-driven programs.